David Kirby 92/95 ECE 438 - Homework 1 Due: 21 February 2020

1. High-level languages use compilers to convert code such as C into Assembly language. From there, the code is sent to an Assembler which further converts it into machine code that can be understood by the computer. While the C code and Assembly language can be human readable, the machine language is not usually decipherable by humans.

2. $$die\ area = 18.9mm \times 13.6mm = 257.04mm^2 = 2.5704cm^2$$

$$wafer\ diameter = 300mm = 30cm$$

(a) $$\frac{dies}{wafer} = \frac{\pi\left(\frac{wafer\ diameter}{2}\right)^2}{die\ area} - \frac{\pi(wafer\ diameter)}{\sqrt{2 \times (die\ area)}} = 233.43 \quad \rightarrow \quad 233\frac{dies}{wafer}$$

(b) $$die\ yield = \frac{wafer\ yield}{\left(1 + \frac{defects}{area} \times die\ area\right)^N} = \frac{0.99}{\left(1 + \frac{0.02}{1cm} \times 2.5704cm^2\right)^7} = 0.6970$$

(c) $cost\ per\ wafer = \$6500 \qquad dies\ per\ wafer = 233\ dies \qquad die\ yield = 0.697$

$$cost\ per\ die = \frac{cost\ per\ wafer}{dies\ per\ wafer \times\ die\ yield} = \frac{\$6500}{233 \times 0.697} = \$40.02\ per\ die$$

(d) $cost\ per\ die = \$40.02 \qquad price\ per\ die = 1.5 \times \$40.02 = \$60.03$

$$wafer\ yield = die\ yield \times dies\ per\ wafer = 233 \times 0.697 = 162$$

$$price\ per\ wafer = wafer\ yield \times price\ per\ die = \$9748.93$$

This is a profit of \$3248.93 per wafer.

(e) $$\#\ failed\ dies \times \%\ repackaged = \#\ repackaged\ dies$$

$$= 71 \times 0.65 = 46\ repackaged\ dies$$

$$\$\ quad-core\ \times \%\ marked\ down = \$\ repackaged\ per\ die$$

$$= \$60.03 \times 0.60 = \$36.02\ per\ repackaged\ die$$

$$\#\ repackaged\ dies \times \$\ per\ repackaged\ die = \$\ per\ wafer\ repackaged$$

$$= 46 \times \$36.02 = \$1656.92\ per\ wafer\ repackaged$$

This repackaged total, with the total from the quad-core, would result in a gross of \$11405.85, which equates to a net profit of \$4905.85 per wafer. If 15% of these gains go to R&D, that would be \$1710.88, leaving \$9694.97 remaining. ??

3. $P1 : clk = 1.8GHz, CPI = 0.8 \qquad P2 : clk = 2.4GHz, CPI = 1.2$

$$P3 : clk = 3.2GHz, CPI = 1.6$$

(a) P3 has the highest clk rate : $3.2GHz$.

$$P1 : \frac{1}{clk} = \frac{1}{1.8GHz} = 0.555ns \qquad P2 : \frac{1}{clk} = \frac{1}{2.4GHz} = 0.417ns$$

$$P3 : \frac{1}{clk} = \frac{1}{3.2GHz} = 0.313ns$$

(b) $MIPS = \frac{Clk\ Cycles}{CPI \times 10^6}$

$$P1 : \frac{1.8 \times 10^9 \frac{cycles}{sec}}{0.8 \times 10^6 \frac{cycles}{instr}} = 2250\ MIPS \qquad P2 : \frac{2.4 \times 10^9 \frac{cycles}{sec}}{1.2 \times 10^6 \frac{cycles}{instr}} = 2000\ MIPS$$

$$P3 : \frac{3.2 \times 10^9 \frac{cycles}{sec}}{1.6 \times 10^6 \frac{cycles}{instr}} = 2000\ MIPS$$

P1 has the highest MIPS rate : 2250 $MIPS$.

(c) $t_{exec} = \frac{instr}{MIPS}$

swap these...

$$P1 : \frac{22 \times 10^6\ instr}{2000 \times 10^6 \frac{instr}{sec}} = 11.0ms \qquad P2 : \frac{19 \times 10^6\ instr}{2250 \times 10^6 \frac{instr}{sec}} = 8.44ms$$

$$P3 : \frac{22 \times 10^6\ instr}{2000 \times 10^6 \frac{instr}{sec}} = 11.0ms$$

P2 has the fastest execution time : 8.4ms.

(d) $speedup = \frac{reference\ time}{processor\ time}$

$$P1 : \frac{35ms}{11ms} = 3.1818 \times \qquad P2 : \frac{35ms}{8.44ms} = 4.1448 \times$$

$$P3 : \frac{35ms}{11ms} = 3.1818 \times$$

4.

(a) The system is I/O bound.

(b) $speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\frac{0.4}{20\times} + 0.6} = 1.6129 \times$

(c) $speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\frac{0.6}{3\times} + 0.4} = 1.667 \times$

(d) We could improve the I/O by increasing performance of the disk access. This can be done by swapping out an older, platter-style HDD with a faster SSD drive, or by choosing a drive that has a faster RPM rating.

5. $t_{exec} = 1000s$

(a) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\frac{0.8}{4\times} + 0.2} = 2.5\times$$

(b) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\frac{0.8}{8\times} + 0.2} = 3.3\times$$

t_{ex} ??

(c) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\cancel{\frac{0.8}{\infty}} + 0.2} = 5\times$$

−1

(d) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\cancel{\frac{0.9}{\infty}} + 0.1} = 10\times$$

6. $t_{exec} = 2s$

(a) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\cancel{\frac{0.95}{\infty}} + 0.05} = 20\times$$

(b) $$speedup_{overall} = \frac{1}{\frac{affected}{speedup_{affected}} + unaffected} = \frac{1}{\frac{0.95}{50\times} + 0.05} = 14.4927\times$$

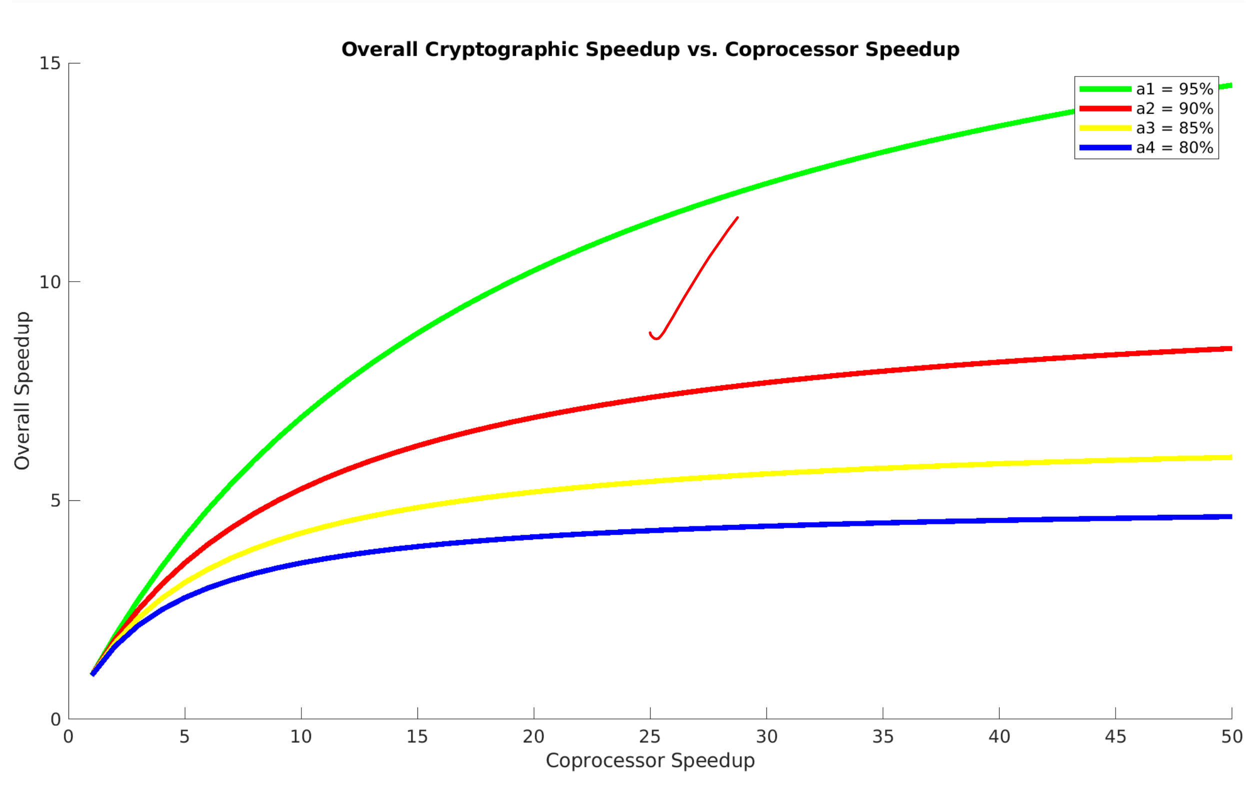
$$t'_{exec} = \frac{t_{exec}}{14.4927\times} = 0.138s$$

(c) $$Energy = Power \times t_{exec} = 2P$$

$$Energy' = 2 \times Power \times t'_{exec} = 0.276P$$

$$\frac{Energy}{Energy'} = \frac{2\cancel{P}}{0.276\cancel{P}} = 7.2464$$

(d) Please see plot below for speedup of the overall cryptographic operation as the coprocessor speedup varies.



MATLAB Drive > ECE438_HW01_P06.m

```
figure(1); clf
coprocessor=1:1:50;
a1=0.95;a2=0.90;a3=0.85;a4=0.80;
overall1=1 ./ ((a1 ./ coprocessor) + (1-a1));
overall2=1 ./ ((a2 ./ coprocessor) + (1-a2));
overall3=1 ./ ((a3 ./ coprocessor) + (1-a3));
overall4=1 ./ ((a4 ./ coprocessor) + (1-a4));
hold
plot(coprocessor,overall1,'g','LineWidth',3)
plot(coprocessor,overall2,'r','LineWidth',3)
plot(coprocessor,overall3,'y','LineWidth',3)
plot(coprocessor,overall4,'b','LineWidth',3)
xlabel('Coprocessor Speedup')
ylabel('Overall Speedup')
title('Overall Cryptographic Speedup vs. Coprocessor Speedup')
legend('a1 = 95%','a2 = 90%','a3 = 85%','a4 = 80%')
```

Very nice!

7. $90nm \qquad f = 500MHz \qquad V = 1.2V \qquad P_D = 1.0W \qquad P_S = 0.2W$

$$P_D \propto \frac{1}{2}CV^2 f \quad \rightarrow \quad C = \frac{2P_D}{V^2 f} = \frac{2W}{(1.2V)^2 \times 500MHz} = 2.778nF$$

$$P_S \propto VI_{leakage} \quad \rightarrow \quad I_{leakage} = \frac{P_S}{V} = \frac{0.2W}{1.2V} = 166.7mA$$

(a) $E = P_{total} \times t_{exec} = \left(P_D + P_S\right) \times t_{exec} = 1.2W \times 1s = 1.2J$

(b) $f = 250MHz \quad V = 1.2V \quad t_{exec} = 2s$

$$P_{total} = P_D + P_S = \frac{1}{2}CV^2 f + VI_{leakage} = 500mW + 200mW = 700mW$$

$$E = P_{total} \times t_{exec} = \left(P_D + P_S\right) \times t_{exec} = 700mW \times 2s = 1.4J$$

(c) $f = 250MHz \quad V = 1.2V \quad t_{exec} = 2s$

$$P_{total} = P_D + P_S = \frac{1}{2}CV^2 f + VI_{leakage} = 281.25mW + 150mW = 431.25mW$$

$$E = P_{total} \times t_{exec} = \left(P_D + P_S\right) \times t_{exec} = 431.25mW \times 2s = 862.5mJ$$

(d) $f = 650MHz \quad V = 1.4V \quad t_{exec} = \frac{500MHz}{650MHz} = 0.769s$

$$P_{total} = P_D + P_S = \frac{1}{2}CV^2 f + VI_{leakage} = 1.7694W + 233.33mW = 2.0028W$$

$$E = P_{total} \times t_{exec} = \left(P_D + P_S\right) \times t_{exec} = 2.0028W \times 0.769s = 1.5406J$$

(e) Other ways to reduce energy could be to increase the size of the die which will decrease the leakage current through the device and decrease the static power consumption.

(f) $\frac{C'}{C} = \frac{130nm}{90nm} = 1.444$

See solutions!

$130nm: \quad P_D \propto \frac{1}{2}CV^2 f = 1.5031W \quad P_S \propto VI_{leakage} = 250mW \quad P_{total} = 1.7531W$

energy per op?

$90nm: \quad P_D \propto \frac{1}{2}CV^2 f = 1.0406W \quad P_S \propto VI_{leakage} = 250mW \quad P_{total} = 1.2906W$

-1

8. (a) & (b) Please see table below for the SPECratio for Machines A and B as well as the geometric means of the SPECratios for both machines. Also shown are the results when libquantum was removed from the calculations.

Benchmark	Ultra 5 Time (sec)	Machine A Time (sec)	Machine A SPEC ratio	Machine B Time (sec)	Machine B SPEC ratio
perlbench	9,770	454	21.520	253	38.617
bzip2	9,650	520	18.558	438	22.032
gcc	8,050	461	17.462	234	34.402
mcf	9,120	268	34.030	150	60.800
gobmk	10,490	429	24.452	383	27.389
hmmer	9,330	197	47.360	135	69.111
sjeng	12,100	529	22.873	362	33.425
libquantum	20,720	91	227.692	7	2960.0
h264ref	22,130	599	36.945	427	51.827
omnetpp	6,250	305	20.492	163	38.344
astar	7,020	327	21.468	234	30.000
xalancbmk	6,900	253	27.273	117	58.974
Geometric mean			30.446		56.982

Benchmark	Ultra 5 Time (sec)	Machine A Time (sec)	Machine A SPEC ratio	Machine B Time (sec)	Machine B SPEC ratio
perlbench	9,770	454	21.520	253	38.617
bzip2	9,650	520	18.558	438	22.032
gcc	8,050	461	17.462	234	34.402
mcf	9,120	268	34.030	150	60.800
gobmk	10,490	429	24.452	383	27.389
hmmer	9,330	197	47.360	135	69.111
sjeng	12,100	529	22.873	362	33.425
h264ref	22,130	599	36.945	427	51.827
omnetpp	6,250	305	20.492	163	38.344
astar	7,020	327	21.468	234	30.000
xalancbmk	6,900	253	27.273	117	58.974
Geometric mean			25.357		39.790

(c) From the results we can see that Machine B is faster (by about 26.5s) than Machine A based off of the geometric mean of the SPECratios of the two machines compared to a relative machine. Speedup of B over A? −1

(d) With libquantum removed from the results, Machine B is still faster than Machine A, but this time by a much smaller margin (14.4s).

9.

(a) $Compiler_A:\quad \#instr = 8\times10^8 \quad t_{exec} = 1.1s \quad clk = 1ns \;\rightarrow\; 1GHz$

$Compiler_B:\quad \#instr = 1.2\times10^9 \quad t_{exec} = 1.6s$

$$CPI_A = \frac{t_{exec}\times f}{\#instr} = \frac{1.1s\times1GHz}{8\times10^8} = 1.375\ cycles\ per\ instruction$$

$$CPI_B = \frac{t_{exec}\times f}{\#instr} = \frac{1.6s\times1GHz}{1.2\times10^9} = 1.333\ cycles\ per\ instruction$$

(b) $Clock\ rate_A = CPI_A \times \#instr_A = 1.1GHz$

$Clock\ rate_B = CPI_B \times \#instr_B = \cancel{1.2GHz} \quad \rightarrow \quad \frac{1.1}{\cancel{1.2}} = \cancel{0.917}$

hmm...

(c) $Compiler_C: \quad \#instr = 7.5 \times 10^8 \quad CPI = 1.3$

$$t_{exec} = \frac{CPI_C \times \#instr}{f} = \frac{7.5 \times 10^8 \times 1.3}{1GHz} = 0.975s$$

$$speedup_{\frac{A}{C}} = \frac{1.1s}{0.975s} = 1.1282\times \qquad speedup_{\frac{B}{C}} = \frac{1.6s}{0.975s} = 1.641\times$$